PATENT APPLICATION

OPTIMIZED TECHNOLOGY MAPPING TECHNIQUES FOR PROGRAMMABLE CIRCUITS

Inventor: Low Yau Loong, a citizen of Malaysia, residing at
J6-1, Taman Bukit Jambul
Bayan Lepas, Penang, 11900 Malaysia

Assignee: Altera Corporation
101 Innovation Drive
San Jose, CA, 95134

Entity: Large

OPTIMIZED TECHNOLOGY MAPPING TECHNIQUES FOR PROGRAMMABLE CIRCUITS

BACKGROUND OF THE INVENTION

[0001]   The present invention relates to techniques for mapping a circuit design to a programmable integrated circuit, and more particularly, to efficient technology mapping techniques for programmable integrated circuits.

[0002]   Programmable integrated circuits can be programmed to implement many types of circuit designs. Examples of programmable integrated circuits include programmable logic devices (PLDs), complex programmable logic devices (CPLDs), field programmable gate arrays (FPGAs), programmable logic arrays (PLAs), and configurable logic arrays (CLAs).

[0003]   Programmable integrated circuits include numerous programmable logic blocks. Each programmable logic block can be individually configured to implement combinatorial or sequential logic operations.

[0004]   Altera's MAX family of PLDs are examples of programmable logic devices. Altera MAX devices include numerous programmable logic blocks called macrocells. Each macrocell has a logic array that can be configured to implement various combinatorial functions. Each macrocell also has a flip-flop that can implement sequential logic functions. Arrays of macrocells are grouped together into logic array blocks (LABs) on a MAX device.

[0005]   A circuit design created by a user can be mapped onto a programmable integrated during a three step process. This process involves synthesis and technology mapping, placement, and routing. During synthesis and technology mapping, logic gates in the user design are converted into programmable logic blocks such as macrocells on a MAX device.

[0006]   Specifically, synthesis is a process that generates an optimized, technology-independent, gate-level network from a user design. Technology mapping is a process that takes the gate-level network generated during synthesis and converts it to a network of programmable logic blocks. A technology mapping process for a MAX device converts a gate-level network into a network of macrocells.

[0007]   A typical technology mapping process for a CPLD consists of two phases, a mapping phase and a macrocell-building phase. The mapping phase determines how the

gate-level network should be converted to a macrocell network. The macrocell-building phase creates the macrocell network based on information provided by the mapping phase. Some tool also provide a third phase to optimize the macrocell network.

[0008]   During the mapping phase, the mapper tool processes all the logic cones in the gate-level network of the user design. For each logic cone, the mapper tool maps nodes in the gate-level network in a topological order starting from the input boundary nodes. The topological ordering guarantees that every node is processed after all of its predecessors have been processed.

[0009]   The mapper tool maps the logic cones by looking for output boundary nodes (i.e. registers or I/O pins). Starting from an output boundary node, the tool traverses backwards to the predecessor nodes in a depth-first-search manner, until reaching the input boundaries before mapping the nodes in a topological order.

[0010]   Altera's Quartus II MAX Technology Mapper uses a multiple-merging-group algorithm during the mapping phase to provide information for the subsequent macrocell-building phase. It processes the nodes in a topological order starting from the input boundary nodes. Each non-boundary node tries to merge with different groups of predecessor nodes to form a macrocell. If all the merging groups fail to form a feasible macrocell, the node itself is formed as a new macrocell.

[0011]   Each different merging group generates different depth and area of a macrocell network for the logic cone. Hence, the Technology Mapper selects the best merging group for the node-under-mapping by using a cost metric. The cost metric consists of depth and area information of macrocell network for the logic cone. Depth cost has priority over area cost to reduce the delay of the macrocell network.

[0012]   The merging groups that the Quartus II Max Technology Mapper uses include (a) merging with all predecessor nodes, (b) merging with nodes until input boundaries or SOFT buffer boundaries, (c) merging with immediate fanin nodes, and (d) merging with nodes included in the best merging group of immediate fanin nodes.

[0013]   In a multiple-merging-group algorithm such as Altera's Quartus II Max Technology Mapper, *every* node chooses among different merging groups to form a macrocell. The major weakness of this algorithm is that it cannot create a minimal depth macrocell network, though it may reduce area utilization.

2

[0014]   Another type of prior art technology mapping tool is called a labeling algorithm. A labeling algorithm is used during the mapping phase of technology mapping to provide information for the subsequent macrocell-building phase. Its primary goal is to minimize the depth of a logic cone by annotating labels on the nodes during the mapping process.

[0015]   In a labeling algorithm, every node has only one merging group, i.e. a node merges with predecessor nodes that have the maximum label. The major weakness for a labeling algorithm is that it does not consider other merging groups that may help in further reducing depth and area utilization of the macrocell network.

[0016]   Therefore, it would be desirable to provide a more robust technology mapping process that optimizes signal delays and reduces area utilization in a synthesized user design for a programmable integrated circuit.

BRIEF SUMMARY OF THE INVENTION

[0017]   The present invention provides technology mapping techniques for user designs of programmable integrated circuits. The techniques of the present invention process the nodes of a user design in a topological order starting from the input boundary nodes.

[0018]   A technology mapping process of the present invention attempts to combine each non-strategic node and predecessor nodes into one programmable logic block according to only one type of node merging group. If the non-strategic node can be feasibly merged with predecessor nodes according to the merging group, they are grouped into a programmable logic block.

[0019]   Other merging groups are considered only when the technology mapping process reaches a strategic node of a logic cone. A strategic node of a logic cone is a node that is the immediate predecessor of the logic cone's output boundary node, and that does not fanout to any other non-boundary nodes. The technology mapping process considers combining a strategic node and predecessor nodes into one programmable logic block according to multiple types of merging groups. If none of these merging groups succeed to form a feasible programmable logic block, the node is formed as a new programmable logic block by itself.

[0020]   If more than one type of merging group can form a feasible programmable logic block, the technology mapping process selects the best merging group for a strategic node by using a cost metric. The cost metric consists of depth and area information of a network for

the logic cone. According to one embodiment, depth cost has priority over area cost to achieve minimal delay.

[0021] Other objects, features, and advantages of the present invention will become apparent upon consideration of the following detailed description and the accompanying drawings, in which like reference designations represent like features throughout the figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0022] Figure 1 illustrates an example of a logic cone that represents a user design for a programmable integrated circuit that can be used with the technology mapping techniques of the present invention;

[0023] Figure 2 illustrates an example of how a technology mapping process of the present invention can attempt to merge a non-strategic node with predecessor nodes that have the maximum label;

[0024] Figure 3A illustrates an example of how a technology mapping process of the present invention can attempt to merge a strategic node with all of its predecessor nodes;

[0025] Figure 3B illustrates an example of how a technology mapping process of the present invention can attempt to merge a strategic node with predecessor nodes until input boundaries or soft buffer boundaries are reached;

[0026] Figure 3C illustrates an example of how a technology mapping process of the present invention can attempt to merge a strategic node with immediate fanin nodes;

[0027] Figure 3D illustrates an example of how a technology mapping process of the present invention can attempt to merge a strategic node with nodes that are included in the best group of immediate fanin nodes;

[0028] Figure 3E illustrates an example of how a technology mapping process of the present invention can attempt to merge a strategic node with predecessor nodes that have the maximum label;

[0029] Figure 3F is a flow chart that illustrates a process for merging non-strategic nodes and strategic nodes with their predecessor nodes into programmable logic blocks according to an embodiment of the present invention;

4

[0030] Figure 4A illustrates an example of a prior art technology mapping process that does not consider various merging groups for a strategic node;

[0031] Figure 4B illustrates an example of a technology mapping process of the present invention that considers various merging groups for a strategic node according to an embodiment of the present invention;

[0032] Figure 4C illustrates an example of a technology mapping process that merges a strategic node with its immediate fanin nodes according to an embodiment of the present invention;

[0033] Figure 5A illustrates an example of a prior art technology mapping process that merges a non-strategic node with its immediate fanin nodes;

[0034] Figure 5B illustrates an example of a technology mapping process of the present invention that considers various merging groups for strategic nodes;

[0035] Figure 5C illustrates another example of a technology mapping process of the present invention that considers various merging groups for strategic nodes;

[0036] Figure 6 illustrates merging constraints for a nodes that are not monotone according to an embodiment of the present invention;

[0037] Figure 7 is a simplified block diagram of a macrocell that can be used with the technology mapping techniques of the present invention;

[0038] Figure 8 is a simplified block diagram of a programmable logic device that can be used with technology mapping techniques of the present invention; and

[0039] Figure 9 is a block diagram of an electronic system that can implement embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0040] The present invention involves technology mapping techniques for mapping logic cones to networks of programmable logic blocks. For example, the present invention can be used to map logic cones to networks of macrocells.

[0041] However, the techniques of the present invention are not limited to macrocells. Macrocells are discussed herein merely as one example of a programmable logic block that

can be used with the present invention. The technology mapping techniques of the present invention can also be used to map logic cones onto other types of programmable logic blocks that are located in different types of PLD and FPGA architectures. For example, the present invention can be used to map logic cones to logic elements, groups of logic elements called logic array blocks (LABs), and configurable logic blocks.

[0042] Figure 1 illustrates an example of a logic cone. A logic cone is a logic region of a user design for a programmable integrated circuit that lies between one output boundary node and several input boundary nodes. In Figure 1, the logic cone lies between input boundary nodes 111-116 and output boundary node 130. Input boundary nodes can include input pins and registers. Output boundary nodes can include output pins and registers. The logic cone includes nodes 117-123.

[0043] A sub-cone is a logic region that lies between one node and all its predecessor nodes, ending at input boundary nodes. A sub-cone lies within a logic cone. In Figure 1, nodes 121 and 117-118 are a sub-cone that lie between node 121 and input boundary nodes 111-113. Predecessor nodes are fanin nodes and are pictured to the left of a node in the Figures. Nodes 117-118 are predecessor nodes to node 121.

[0044] A non-strategic node is a non-boundary node in a logic cone that is not a strategic node. A strategic node of a logic cone is a node that is the immediate predecessor of the logic cone's output boundary node, and that does not fanout to any other non-boundary nodes. The non-strategic nodes in the Figure 1 are nodes 117-122. Node 123 is a strategic node, because it fans out to output boundary node 130.

[0045] The technology mapping process of the present invention attempts to merge each non-strategic node and its predecessor nodes that have a maximum label, as shown, for example in Figure 2. In Figure 2, the input boundary nodes are nodes 211-218, and the non-boundary, non-strategic nodes are 219-229.

[0046] A technology mapping process of the present invention processes each node within a logic cone in a topological order starting from the input boundary nodes and ending at an output boundary node. The process considers whether each non-boundary node can be merged with any of its predecessor nodes into one macrocell.

[0047] The technology mapping process assigns a label to each node in the logic cone. Initially, each input boundary node is assigned the label 0. The technology mapping process

then attempts to combine each non-strategic node with its predecessor nodes that have the maximum label value. The process first determines if the combination is feasible given the inherent constraints of the macrocell architecture. If the combination is feasible, the non-strategic node is combined with its predecessor nodes that have the maximum label value into one macrocell, and then the non-strategic node is assigned that maximum label value.

[0048] For example, node 227 in Figure 2 is considered to be the current node-under-mapping. Node 224 has the maximum label value (2) of all of the predecessor nodes of node 227. Therefore, if nodes 227 and 224 can be feasibly merged into one macrocell, node 227 is assigned the label 2, and nodes 224 and 227 are combined into one macrocell. If nodes 224, 227 and 229 can be feasibly merged into one macrocell, node 229 is assigned a label of 2, and nodes 224 , 227 and 229 are merged into one macrocell. Node 229 is a non-strategic node if its output is not coupled directly to an output boundary node.

[0049] The technology mapping process of the present invention only combines nodes into one macrocell after first determining whether the combination is feasible given the architecture of the macrocell. After a particular group of nodes has been selected as a possible merging group, the merging group is converted into another logic form with similar functionality.

[0050] The converted logic is in the form of (k, m, 1), which represents with k inputs, m product-terms, and 1 output. This logic form is also called Sum-Of-Product (SOP). A feasible macrocell means that the converted SOP form must have a number of inputs <= k, a number of product-terms <= m, and no more than 1 output.

[0051] This feasibility determination method can be applied to the process of determining whether any merging group of nodes discussed herein can be feasibly merged into one macrocell. Feasibility determination methods are different for other types of programmable logic block architectures. Feasibility determinations for Altera's MAX family of PLDs are well known.

[0052] If a non-strategic node cannot be feasibly merged into one macrocell with its predecessor nodes that have the maximum label given the macrocell architecture, the non-strategic node is assigned the maximum label of its predecessor nodes plus 1. For example, nodes 219-223 are assigned a label of 1, because they cannot be merged with an input boundary node. Therefore, each of nodes 219-223 are assigned a label of 0 + 1 = 1.

7

[0053]   According to the present invention, the technology mapping process only merges non-strategic nodes into a macrocell with predecessor nodes according to the labeling process previously discussed. The technology mapping process does not consider other merging groups for non-strategic nodes.

[0054]   The technology mapping process of the present invention tries to merge each strategic node with two or more merging groups of predecessor nodes to form a new macrocell. If none of the merging groups that are considered can be combined to form a feasible macrocell, the strategic node is formed as a new macrocell by itself.

[0055]   Figures 3A-3E illustrate examples of five different node merging groups that can be considered for combining a strategic node with one or more of its predecessor nodes. The various merging groups shown in these Figures are only considered when a strategic node is the node-under-mapping. Each merging group may generate a different depth and area of a macrocell network for a logic cone.

[0056]   A technology mapping process of the present invention process can select the best merging group for the strategic node by using a cost metric. The cost metric can consist of depth and area information of macrocell network for the logic cone.

[0057]   Depth cost is based on the maximum number of macrocell stages from an input boundary to the node-under-mapping. Area cost indicates the total number of macrocells in a sub-cone that of the node-under-mapping. In general, depth cost has priority over area cost to achieve minimal delay macrocell network, and thus depth cost is compared first. If both depth costs are the same for two or more possible merging groups, then only area costs are compared.

[0058]   A technology mapping process of the present invention considers various types of merging groups for combining a strategic node with one or more of its predecessor nodes into one macrocell. One type of merging group combines a strategic node with all of its predecessor non-boundary nodes. Figure 3A illustrates an example of this type of merging group.

[0059]   In Figure 3A, strategic node 123 is merged with all of its predecessor nodes 117-122 that are not input boundary nodes. The merged group is shown inside the dotted oval in Figure 3A.

8

[0060] The technology mapping process of the present invention then determines whether all of nodes 117-123 can be feasibly merged into one macrocell given the constraints of the macrocell architecture. If the nodes can be feasibly merged into a macrocell, the process calculates a depth cost and an area cost for the macrocell network formed by the merger. If nodes 117-123 are all merged into one macrocell, the depth cost equals one, because there would be only one macrocell between the input and output boundary nodes. The area cost also equals one, because the sub-cone is reduced to one macrocell.

[0061] Another merging group that can be used to merge a strategic node with one or more of its predecessor nodes is illustrated in Figure 3B. In this type of merging group, a strategic node is combined with its predecessor nodes in all directions until an input boundary node or a soft buffer boundary node is reached. A soft buffer node is a dummy node that marks a boundary.

[0062] In Figure 3B, strategic node 123 is merged with predecessor nodes 117-119 and 121-122. To form this merging group, the technology mapping process traverses each fanin of strategic node 123. Traversing each of the fanins to node 123 leads to nodes 121, 117 and 118 as well as nodes 122 and 119. The process terminates at input boundary nodes 111-114. Unlike in the example of Figure 3A, the process also terminates at soft buffer boundary node 125. The resulting merging group is shown inside the dotted oval in Figure 3B.

[0063] The technology mapping process then determines whether nodes 117-119 and 121-123 can be feasibly merged into one macrocell. If nodes 117-119 and 121-123 can be feasibly merged into a macrocell, the process calculates a depth cost and an area cost for the macrocell network formed by the merged nodes.

[0064] If nodes 117-119 and 121-123 are merged into one macrocell, the depth cost equals two. Because node 120 forms a separate macrocell, there is a maximum of two stages of macrocells between the input boundary nodes and the output boundary nodes. The area cost also equals two, because two separate macrocells are formed.

[0065] Another merging group that can be used to merge a strategic node with one or more of its predecessor nodes is illustrated in Figure 3C. In this type of merging group, a strategic node is combined only with its immediately preceding fanin nodes.

[0066] In Figure 3C, strategic node 123 is merged only with its immediately preceding fanin nodes 121-122. Unlike the merging groups in the preceding examples, the technology

mapping process only merges the strategic node with fanin nodes that are directly connected to inputs of the strategic node as shown in Figure 3C.

[0067]   The technology mapping process then determines whether nodes 121-123 can be feasibly merged into one macrocell. If nodes 121-123 can be feasibly merged into a macrocell, the process calculates a depth cost and an area cost for the macrocell network formed by the merged nodes. If nodes 121-123 are merged into one macrocell, the depth cost equals two, because there is a maximum of two levels of macrocells between the input nodes and the output node. The area cost equals five, because nodes 117-120 each form a separate macrocell.

[0068]   Another merging group that can be used to merge a strategic node with one or more of its predecessor nodes is illustrated in Figure 3D. In this type of merging group, a strategic node is combined with the best merging groups of the strategic node's immediate fanin nodes.

[0069]   In Figure 3D, strategic node 123 is merged with its immediately preceding fanin nodes 121-122 and each of their best merging groups into group 161. The best merging groups for nodes 121 and 122 are identified by using the maximum label technique for non-strategic nodes discussed above with respect to Figure 2. Thus, the best merging group for a non-strategic node (e.g., for nodes 121-122) is either the non-strategic node by itself or the non-strategic node combined with its predecessor nodes that have the maximum label.

[0070]   The best merging group for node 121 is group 162, which includes only node 121. The best merging group of node 122 includes is group 163. Group 163 includes nodes 119-120 and 122. Thus, node 122 can be feasibly merged with its two predecessor nodes that have the maximum label (1), but node 121 cannot be merged with nodes 117-118, due to limitations of the macrocell architecture.

[0071]   The technology mapping process then determines whether nodes 119-123 can be feasibly merged into one macrocell. If nodes 119-123 can be feasibly merged into a macrocell, the process calculates a depth cost and an area cost for the macrocell network formed by the merged nodes. If nodes 119-123 are merged into one macrocell, the depth cost equals two, because there is a maximum of two levels of macrocells between the input nodes and the output node. The area cost equals three, because nodes 117-118 each form a separate macrocell.

[0072]   Another merging group that can be used to merge a strategic node with one or more of its predecessor nodes is illustrated in Figure 3E. In this type of merging group, a strategic node is combined with its predecessor nodes that have the maximum label. This merging group is discussed in detail above with respect to non-strategic nodes in Figure 2. In Figure 3E, strategic node 123 is merged only with node 121, because node 121 has the maximum label (2) among the predecessor nodes of node 123.

[0073]   If nodes 123 and 121 are merged into one macrocell, the depth cost equals two. Because nodes 117-118 form separate macrocells, and nodes 119, 120, 122 are merged into one macrocell, there is a maximum of two stages of macrocells between the input boundary nodes and the output boundary nodes. The area cost also equals four, because four separate macrocells are formed.

[0074]   In the present invention, non-strategic nodes can only be merging with predecessor nodes that have the maximum label. However, strategic nodes can be merged with predecessor nodes according to any of the merging groups shown and discussed with respect to Figures 3A-3E.

[0075]   If the nodes in the merging group of Figure 3A can be feasibly merged into one macrocell, the technology mapping process selects this merging group to combine nodes 117-123, because it has the lowest depth cost of the five possible merging groups. If the merging group of Figure 3A cannot be feasibly merged into one macrocell, the process selects the merging group of Figure 3B, because this merging group has a depth cost of 2 and an area cost of 2 which is the second best alternative.

[0076]   If the nodes in the Figure 3B merging group cannot be feasibly merged into one macrocell, the technology mapping process selects the merging group of Figure 3D. If the nodes in the Figure 3D merging group cannot be feasibly merged into one macrocell, the process selects the merging group of Figure 3E. The merging group of Figure 3C is the last choice. If the nodes cannot be feasibly combined using any of the five merging groups, the strategic node is formed as a new macrocell by itself.

[0077]   The technology mapping process of the present invention can consider merging a strategic node with less than the five merging groups illustrated in Figures 3A-3E. Also, the technology mapping process of the present invention can consider other merging groups in addition to the five merging groups shown in Figures 3A-3E.

11

[0078] Figure 3F illustrates a generalized process for combining nodes in a logic cone into programmable logic blocks such as macrocells and logic elements according to an embodiment of the present invention. At step 301, a technology mapping tool combines non-strategic nodes in a logic cone with predecessor nodes into programmable logic blocks according to only one type of node merging group.

[0079] Figure 2 illustrates one example of a node merging group that can be used for combining non-strategic nodes as discussed above. Nodes in a logic cone are combined together into a programmable logic block only if the combination is feasible given the inherent limitations of a particular programmable logic block architecture. For example, a macrocell may be limited to 3 input signals. In this example, nodes cannot be combined into one macrocell if together they have more than 3 input signals.

[0080] At step 302, the technology mapping tool considers the feasibility of combining a strategic node with one or more of its predecessor nodes into a programmable logic block by considering multiple types of node merging groups. The feasibility of combining nodes into one programmable logic block is determined by the architecture of the programmable logic block, as previously discussed. Figures 3A-3E illustrate examples of various types of node merging groups that can be used for combining a strategic node with its predecessor nodes as discussed above.

[0081] At step 303, the technology mapping tool selects one of the node merging groups for the strategic node based on a cost metric. The cost metric can be based on the area and/or the depth of the resulting programmable logic block architecture. For example, the tool may select the merging group that generates the macrocell network with the shortest depth.

[0082] At step 304, the technology mapping tool combines the strategic node and one or more of its predecessor nodes into a programmable logic block using the node merging group selected in step 303.

[0083] Figures 4A-4C illustrate how using various merging groups for strategic nodes in a logic cone can greatly effect how a technology mapping process generates a macrocell network. The logic cone shown on the left sides of Figures 4A-4C includes three input boundary nodes 401-403, non-strategic nodes 404-407, strategic node 408, and output node 409.

12

[0084] A prior art technology mapping process does not consider various merging groups for strategic nodes. Such a technology mapping process combines a strategic node 408 with its predecessor nodes that have the maximum label. Figure 4A illustrates how a prior art technology mapping process assigns node 408 with the maximum label (2) of its predecessor nodes (node 406), and then combines nodes 406 and 408 into one macrocell 417. Each of the other nodes 404-405 and 407 are placed in separate macrocells 414-416, respectively.

[0085] A technology mapping process of the present invention considers various merging groups for strategic nodes in a logic cone, as discussed above. Figures 4B and 4C illustrate examples of how the merging groups shown in Figures 3A and 3C can be used to produce a macrocell network.

[0086] A technology mapping process of the present invention can attempt to combine strategic node 408 with all of its predecessor non-boundary nodes 404-407 as discussed above with respect to Figure 3A. If such as combination is feasible, the process combines all of the non-boundary nodes 404-408 into one macrocell 424 as shown in Figure 4B.

[0087] A technology mapping process of the present invention also considers other possible merging groups. For example, the technology mapping process can attempt to combine strategic node 408 with its immediate predecessor nodes 406-407 as discussed above with respect to Figure 3C. If such a combination of nodes is feasible, the technology mapping process can combine nodes 406-408 into one macrocell 436 as shown in Figure 4C. Node 404 is placed in macrocell 434 by itself, and node 405 is placed into macrocell 435 by itself.

[0088] The macrocell network created by the prior art technology mapping process has 4 macrocells grouped into 2 stages as shown in Figure 4A. On the other hand, the macrocell network created by a technology mapping process of the present invention can, for example, have only 1 macrocells as shown in Figure 4B, or 3 macrocells grouped into 2 stages as shown in Figure 4C.

[0089] If the combination of shown in Figure 4B can be feasibly placed into one macrocell network, then a technology mapping process of the present invention selects this combination. If the Figure 4B combination cannot be feasibly placed into one macrocell, then the technology mapping process selects the Figure 4C combination (or some other node combination) that can be feasibly placed into a macrocell network.

13

[0090] In both the Figure 4B and the Figure 4C examples, the present invention reduces the number of macrocells and/or macrocell stages in the resulting macrocell network relative to the prior art technique. Thus, the technology mapping process of the present invention can map a user design onto a more optimized network of macrocells. A more optimized macrocell network has reduced signal delay times.

[0091] Figures 5A-5C illustrate how the present invention can substantially reduce the number of macrocells and macrocell stages (i.e., depth) generated for a logic cone. Figure 5A illustrates the disadvantages of using a prior art technology mapping technique called the Quartus II MAX Technology Mapper. The Quartus II MAX Technology Mapper considers various merging groups for all the nodes in a logic cone (including non-strategic nodes).

[0092] The Quartus II MAX Technology Mapper tries up to 4 merging groups for all of the nodes. These merging groups are (a) merging with all predecessor nodes, (b) merging with predecessor nodes until input boundaries or SOFT buffer boundaries are reached, (c) merging with immediate fanin nodes, and (d) merging with nodes included in the best merging group of immediate fanin nodes.

[0093] The logic cone shown on the left side of Figure 5A includes four input boundary nodes 501-504, non-strategic nodes 505-510, strategic node 511, and output boundary node 512. For node 507, none of the four merging groups is feasible. Therefore, node 507 is placed into a macrocell by itself. For node 509, merging groups (a) and (b) are not feasible. Group (c) is chosen as the best group, which includes nodes 507-509, as shown by the dotted oval in Figure 5A. Group (d) is same as group (c) for node 509.

[0094] In the example of Figures 5A merging groups (a), (b) and (d) are not feasible for node 511, so the Quartus II MAX Technology Mapper places node 511 into a macrocell 529 by itself. Merging group (d) is not feasible for node 511, because the best merging group for node 509 is too big (includes nodes 507-509). Merging group (c) is feasible for node 511. However, the macrocell network resulting from merging group (c) has the same cost as the macrocell network shown in Figure 5A, i.e., 5 macrocells with 3 depths.

[0095] According to the example of Figure 5A, the Quartus II MAX Technology Mapper combines node 509 with its immediately preceding input nodes 507-508 to form one macrocell 528. Nodes 505, 506, and 510 are placed into separate macrocells 525, 526, and 527, respectively. Node 511 is placed a macrocell 529 by itself.

14

[0096]   Figure 5B illustrates how a technology mapping process of the present invention can convert logic cone 501-512 into a smaller macrocell network than the prior art technique. In Figure 5B, a technology mapping process of the present invention attempts to combine non-strategic nodes 505-510 using only the merging group of Figure 2 (predecessor nodes with the maximum label).

[0097]   Using this merging group, nodes 507 and 509 are chosen as the best merging group for node 509, because node 507 has the maximum label of 2, and the combination is feasible. For strategic node 511, all of the 5 merging groups discussed above with respect to Figures 3A-3E are considered. The merging groups of Figures 3A and 3B are not feasible.

[0098]   If the merging group of Figure 3D is feasible for node 511, nodes 507, 509, 510, and 511 are combined into one macrocell 550. Feasibility of a combination is determined based on the macrocell architecture. A macrocell network of 4 macrocells with 2 depths is generated, as shown on the right side of Figure 5B. Nodes 505, 506, and 508 are placed into separate macrocells 541-543, respectively.

[0099]   If nodes 507, 509, 510, and 511 cannot feasibly be merged into one macrocell, the present invention determines whether nodes 507, 509, and 511 can be combined into one macrocell according to the merging group of Figure 3E. If this combination is feasible, nodes 507, 509 and 511 are combined into macrocell 539 as shown in Figure 5C. Nodes 505, 506, 508 and 510 are each placed into separate macrocells 535-538, respectively.

[0100]   Thus, the present invention converts logic cone 505-511 into a macrocell network that has a shorter depth and/or less macrocells than the prior art technique. The macrocell network of Figure 5B has 4 macrocells 541-543 and 550 and a depth of 2. The macrocell network of Figure 5C has 5 macrocells 535-539 and a depth of 2. On the other hand, the macrocell network of Figure 5A has 5 macrocells 525-529 and has a depth of 3.

[0101]   It is feasible for a technology mapping process of the present invention to try various merging groups for different types of programmable integrated circuits. For example, (k, m, p)-macrocell based complex programmable logic device (CPLD) have a non-monotone property of merging constraints. Thus, the fact that a merging group cannot be feasibly grouped into one type of macrocell does not imply that a larger merging group including more nodes cannot be feasibly merged into that same type of macrocell.

[0102] This principle is illustrated in Figure 6. Assume that a PLD has macrocells that are limited to only 3 inputs. For this type of architecture, nodes 603-605 in Figure 6 cannot feasibly be merged into one macrocell by themselves, because they have 4 unique input signals. However, if the merging group were expanded to include node 602, then nodes 602-

5   605 can be feasibly merged into macrocell, because the resulting macrocell has only 3 unique input signals.

[0103] Thus, the present invention is advantageous, in part, because it considers multiple types of merging groups for the critical strategic nodes without being locked into considering only one type of merging group. The present invention does not assume that larger merging

10   groups are not feasible for strategic nodes, based on a determination that other types of merging groups are not feasible. The present invention also limits non-strategic nodes to one type of merging group so that the most flexible merging group options are available to the strategic nodes.

[0104] Another advantage of the present invention is that it allows new merging groups to

15   be introduced in the future if the merging group produces good results. Thus, the present invention is not limited to the merging groups discussed herein. Other types of merging groups can be used to combine strategic nodes with one or more of their predecessor nodes into a macrocell.

[0105] The present invention allows delay-critical nodes to be merged as a macrocell.

20   Delay-critical nodes are nodes that lie along the longest delay path of a logic cone. In order to create a minimal depth macrocell network for the logic cone, it is important to merge all delay-critical nodes to form a minimum number of macrocells in a macrocell network.

[0106] In the prior Quartus II MAX Technology Mapper, every node is likely to include non-delay-critical nodes to form a macrocell, which makes the macrocell larger in terms of

25   product term (PTERM) usage. As a result, a successor node has lower chance to merge with a predecessor node.

[0107] The present invention also allows the best mapping solution to be selected from more choices of merging groups for the predecessor of a logic cone's output boundary node (i.e., the strategic node). As a result, the delay of the mapped macrocell network can be

30   improved further, without increasing area utilization.

[0108] Figure 7 illustrates one type of macrocell architecture that can be used according to the principles of the present invention. The macrocell architecture is used in Altera's MAX family of CPLDs. The Figure 7 macrocell can be individually configured for either sequential or combinatorial logic operations. The macrocell of Figure 7 includes three

5 functional blocks: the logic array 701, the product-term select matrix 702, and the programmable register 703.

[0109] Combinatorial logic is implemented in the logic array 701, which is represented by the AND gates in Figure 7. Logic array 701 provides product terms in the macrocell. The product-term select matrix 702 allocates these product terms for use as either primary logic

10 inputs (to the OR and XOR gates) to implement combinatorial functions, or as secondary inputs to the macrocell's register preset, clock, and clock enable control functions. The programmable register 703 implements sequential functions.

[0110] Figure 8 is a simplified partial block diagram of an exemplary high-density PLD 800. PLD 800 embodies another type of PLD architecture that can be implemented with the

15 technology mapping techniques of the present invention. PLD 800 includes a two-dimensional array of programmable logic array blocks (or LABs) 802 that are interconnected by a network of column and row interconnects of varying length and speed. LABs 802 include multiple (e.g., 10) logic elements (or LEs), an LE being a small unit of logic that provides for efficient implementation of user defined logic functions.

20 **[0111]** PLD 800 also includes a distributed memory structure including RAM blocks of varying sizes provided throughout the array. The RAM blocks include, for example, 512 bit blocks 804, 4K blocks 806 and a MegaBlock 808 providing 512K bits of RAM. These memory blocks can also include shift registers and FIFO buffers. PLD 800 further includes digital signal processing (DSP) blocks 810 that can implement, for example, multipliers with

25 add or subtract features. I/O elements (IOEs) 812 located, in this example, around the periphery of the device support numerous single-ended and differential I/O standards. It is to be understood that PLD 800 is described herein for illustrative purposes only and that the present invention can be implemented in many different types of PLDs, FPGAs, and the like.

[0112] While PLDs of the type shown in Figure 8 provide many of the resources required

30 to implement system level solutions, the present invention can also benefit systems wherein a PLD is one of several components. Figure 9 shows a block diagram of an exemplary digital system 900, within which the present invention can be embodied. System 900 can be a

17

programmed digital computer system, digital signal processing system, specialized digital switching network, or other processing system. Moreover, such systems can be designed for a wide variety of applications such as telecommunications systems, automotive systems, control systems, consumer electronics, personal computers, Internet communications and

5   networking, and others. Further, system 900 can be provided on a single board, on multiple boards, or within multiple enclosures.

[0113]   System 900 includes a processing unit 902, a memory unit 904 and an I/O unit 906 interconnected together by one or more buses. According to this exemplary embodiment, a programmable logic device (PLD) 908 is embedded in processing unit 902. PLD 908 can

10   serve many different purposes within the system in Figure 4. PLD 908 can, for example, be a logical building block of processing unit 902, supporting its internal and external operations. PLD 908 is programmed to implement the logical functions necessary to carry on its particular role in system operation. PLD 908 can be specially coupled to memory 904 through connection 910 and to I/O unit 906 through connection 912.

15   [0114]   Processing unit 902 can direct data to an appropriate system component for processing or storage, execute a program stored in memory 904 or receive and transmit data via I/O unit 906, or other similar function. Processing unit 902 can be a central processing unit (CPU), microprocessor, floating point coprocessor, graphics coprocessor, hardware controller, microcontroller, programmable logic device programmed for use as a controller,

20   network controller, and the like. Furthermore, in many embodiments, there is often no need for a CPU.

[0115]   For example, instead of a CPU, one or more PLDs 908 can control the logical operations of the system. In an embodiment, PLD 908 acts as a reconfigurable processor, which can be reprogrammed as needed to handle a particular computing task. Alternately,

25   programmable logic device 908 can itself include an embedded microprocessor. Memory unit 904 can be a random access memory (RAM), read only memory (ROM), fixed or flexible disk media, PC Card flash disk memory, tape, or any other storage means, or any combination of these storage means.

[0116]   While the present invention has been described herein with reference to particular

30   embodiments thereof, a latitude of modification, various changes, and substitutions are intended in the present invention. In some instances, features of the invention can be employed without a corresponding use of other features, without departing from the scope of

the invention as set forth. Therefore, many modifications may be made to adapt a particular configuration or method disclosed, without departing from the essential scope and spirit of the present invention. It is intended that the invention not be limited to the particular embodiments disclosed, but that the invention will include all embodiments and equivalents

5 falling within the scope of the claims.